

Lab 4: Implementing WebAPI Using JavaScript

This lab will guide you in building a WebAPI to search a superhero database using JavaScript. (Note: this is an individual lab, everyone is going to make their own submission)

Environment Setup

Ensure Node.js is installed on your system. You can download the latest version from: <https://nodejs.org/>.

Mac Users: If you're using a Mac, it may be convenient to install Node.js via [Homebrew](#):

```
# Make sure Homebrew is already installed (visit https://brew.sh/ for
instructions).
brew install node@22

# Verify Node.js installation:
node -v # Expected output: `v22.11.0`

# Verify npm installation:
npm -v # Expected output: `10.9.0`
```

Student Code

Download the student code from BB. Extract it and you will find the following files:

```
student_code_api
├── index.js
├── package-lock.json
├── package.json
├── superheroes.csv
└── public
    ├── index.html
    └── main.css
```

,where index.js contains the server JS code, index.html contains the client JS code, and superheroes.csv is the database we are searching from.

Before changing any code you will need to run `npm install` to install any dependencies for this lab. After running this, you should see a new folder called `node_modules`.

Now when you run this API with `npm start` under the project folder, you will see the message in your terminal:

```
> superheroapi@1.0.0 start
> echo 'Start me up!' && node index.js

'Start me up!'
I've become aware.
```

And then, open your web browser, type in `http://localhost:3000/` to open the client webpage. Now your Superhero Searchifier does not do because the client and serve code are not complete.

You can then open `index.js` and `index.html` to complete the code. Similar to Lab 3, follow the comments and complete where the comments says "TODO". You can debug your code by pressing F12 in your web browser and go to tab "console". `Console.log()` function in your code will print to this tab.

After completing the TODO tasks, open `http://localhost:3000/` again and test by searching any superhero name you should see some output like this:

Superhero Searchifier

Enter a search term:

- Peter Parker (Guardian) (Earth-616)
- Peter Parker (Spidercide) (Earth-616)
- Peter Parker (Kaine) (Earth-616)
- Peter Parker (Skrull) (Earth-616)
- Peter Parker (Counter-Earth) (Earth-616)
- Peter Parker (Ben Reilly) (Earth-616)
- Peter Parker (Robot) (Earth-616)
- Peter Parker (Spidercide) (Earth-616)
- Spider-Man (Peter Parker)
- Peter Parker (Ben Reilly) (Earth-616)
- Peter Parker (Guardian) (Earth-616)
- Peter Parker (Doppelganger) (Earth-616)
- Peter Parker (Jack) (Earth-616)
- Peter Parker (Spider-Skeleton) (Earth-616)
- Peter Parker (Kaine) (Earth-616)
- Peter Parker (Robot) (Earth-616)
- Peter Parker (Counter-Earth) (Earth-616)
- Spider-Man (Peter Parker)
- Peter Parker (Skrull) (Earth-616)
- Peter Parker (Spider-Skeleton) (Earth-616)
- Peter Parker (Doppelganger) (Earth-616)
- Bambi (Peter Parker's neighbor) (Earth-616)
- Peter Parker (Jack) (Earth-616)
- Bambi (Peter Parker's neighbor) (Earth-616)

Submission

Archive all original files from the `student_code_api` folder into a compressed file (e.g., ZIP). DO **NOT** include `node_modules` created by Node.js.

Submit the archive through Blackboard.

Rubrics

- 4 points for completing server code in index.js.
- 4 points for completing client code in index.html.
- 2 points for successful search results shown in the webpage without error.